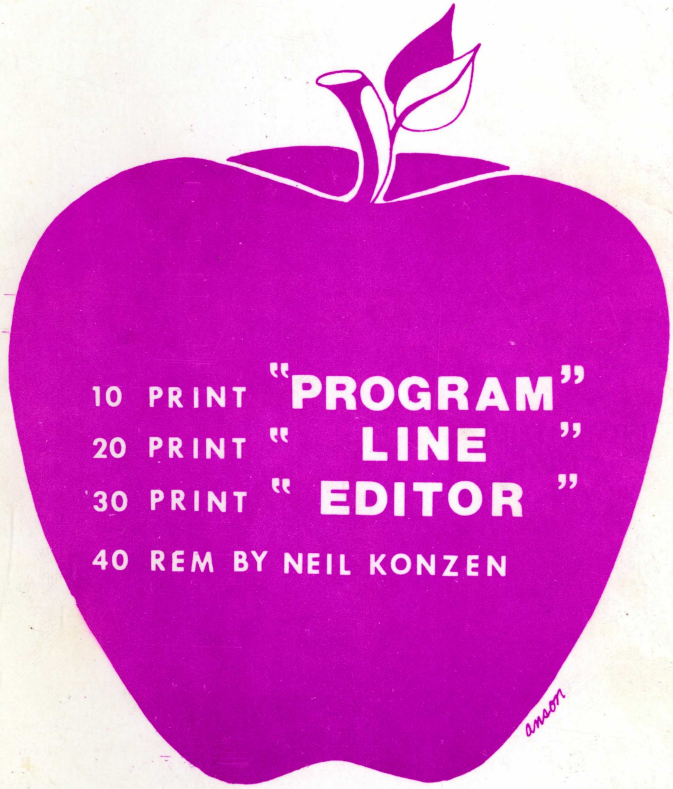
SYNERGISTIC SOFTWARE



PROGRAM LINE EDITOR

BY

NEIL KONZEN

Distributed by
Synergistic Software
5221 120th Ave. S.E.
Bellevue, WA 98006

PROGRAM LINE EDITOR

BY

NEIL KONZEN

PROGRAM LINE EDITOR
SUMMARY OF COMMANDS

SYSTEM COMMANDS

Ctrl-E	Edit BASIC line. Must be followed by a line number or a period (.).
Ctrl-W	Edit either last line typed or line being typed.
Ctrl-Y	Return to BASIC from Monitor.
CALL 1013	Reconnect Editor after use of PR# and IN# (Integer BASIC & Applesoft).
&	Reconnect Editor after use of PR# and IN# (Applesoft only).

EDIT MODE COMMANDS

Ctrl-I	INSERT	Insert character(s) at cursor position.
Ctrl-D	DELETE	Delete character at cursor position.
Ctrl-F	FIND	When followed by a character typed N times, Ctrl-F moves cursor to Nth occurence.
Ctrl-Z	ZAP	When followed by a character typed N times, Ctrl-Z deletes all characters up to Nth occurence of typed character.
Ctrl-O	OVERRIDE	Insert control character at cursor position.
Ctrl-R	RESTART	Re-edit original line.
Ctrl-P	PACK	Remove all spaces from line except those within quotes.
Ctrl-C	CONVERT	Convert character at cursor to case specified by Ctrl-A/Ctrl-S shift lock and advance the cursor.
Ctrl-B	BEGIN	Move cursor to beginning of line.
Ctrl-N	END	Move cursor to end of line.
Ctrl-Q	QUIT	Enter line up to cursor and exit Edit mode.

RETURN		Enter entire line regardless of cursor position and exit Edit mode.
Ctrl-X	CANCEL	Cancel line and exit Edit mode.
<-, ->		Move cursor one space backward and forward, respectively.
Ctrl-A,	Ctrl-S	Upper and lower case shift locks, respectively.

LIST CONTROL COMMANDS

Ctrl-S	STOP	Temporarily stops listings when typed during character output.
Ctrl-C	CANCEL	Terminates listing and returns to BASIC when typed during character output.
Ctrl-F	FLUSH	Disables output to screen when typed during character output.

CASE COMMANDS

Ctrl-A	Upper case shift lock.
Ctrl-S	Lower case shift lock.

PROGRAMMABLE CURSOR CHARACTERS

Ctrl-H	LEFT	Moves cursor one space left when printed.
Ctrl-I	RIGHT	Moves cursor one space right when printed.
Ctrl-J	DOWN	Moves cursor down one line when printed.
Ctrl-K	UP	Moves cursor up one line when printed.

TABLE OF CONTENTS

SECTION I. OVERVIEW

The Program Line Editor is designed to make using the Apple][many times easier and more productive. The many powerful features of the Program Line Editor eliminate much of the tedium and frustration of program editing and development.

The next five sections of this manual deal with the operation of the Program Line Editor. The five primary features are: The Program Line Editor itself, which allows you to modify program lines; the Escape Functions, which enable you to define and execute any sequence of characters with just two keystrokes; the Listing Control feature, which allows you to stop and start listings with the touch of a key; the direct input of Lower case characters; and Programmable cursor control.

Since the Program Line Editor is intended to be used as the greeting program on your own diskettes, Section VIII deals with creating and saving your own Program Line Editor Greeting programs.

Other operating considerations such as re-entering BASIC from monitor and reconnecting the Editor after using the PR# and IN# commands are discussed in Section IX. Memory usage and other information helpful to advanced programmers can be found in Section X.

SECTION II. GETTING STARTED

LOADING AND RUNNING THE PROGRAM LINE EDITOR

APPLE][SYSTEMS WITH INTEGER BASIC

Just insert the included disk in your drive and boot it normally. After a few seconds, a disk CATALOG will appear. When the cursor returns, the Program Line Editor is up and running, ready for your command.

APPLE][PLUS SYSTEMS WITHOUT INTEGER BASIC

If you own an Apple][plus system that does not have an Integer BASIC Firmware card, the Program Line Editor will not be run when the supplied disk is booted. This is because the Greeting program is an Integer BASIC program. So, the following steps must be taken in order to allow the Program Line Editor disk to boot properly in Applesoft:

1. Remove the Write Protect tabs from the edge of the included disk.
2. Insert and boot the disk.
3. UNLOCK PROGRAM LINE EDITOR
4. UNLOCK PLE.FP
5. RENAME PROGRAM LINE EDITOR, PLE.INT
6. RENAME PLE.FP, PROGRAM LINE EDITOR
7. LOCK PROGRAM LINE EDITOR
8. LOCK PLE.INT
9. Replace the Write Protect tab.
10. Reboot the disk.

After this is done, just boot the disk normally. Soon, a disk catalog will appear, letting you know that the Program Line Editor is up and running.

When the Program Line Editor has been run, you shouldn't notice anything unusual about the operation of your Apple. In fact, everything IS the same, except for all of the new commands and features described in the following pages.

NOTE: When the Program Line Editor is up and running, DOS will set HIMEM: 1536 bytes lower than normal for your system. See Section X, GENERAL INFORMATION for more information on memory use by the Line Editor.

SECTION III. PROGRAM LINE EDITING

First we'll take a look at the foremost feature of the Program Line Editor (Program Line Editing, of course). You'll probably want to type in a short BASIC program so you can try out the commands as you read the description of each. As they say, practice makes perfect.

There are two new commands added to BASIC that allow you to edit either a program line or the last line of characters typed at the keyboard. These are Ctrl-E (EDIT) and Ctrl-W.

ENTERING EDIT MODE: CTRL-E AND CTRL-W

CTRL-E

This is the command used when you want to edit a line in your program. To type Ctrl-E, just press the CTRL key and hold it down while you hit the E key. After the word "EDIT" appears, type the line number of the line you want to edit. The line will appear, with the cursor at the beginning of the first statement in the line. The Line Editor remembers the number of the last line edited, so, if you want to edit the same line again later, simply type Ctrl-E and a period (.). This will cause the last- edited line to reappear.

Here are a few things to watch for when using Ctrl-E to enter Edit mode:

1. Ctrl-E must be the first character typed on a line.
2. Don't try to edit line 0 of an Applesoft program.
3. Ctrl-E is disabled during INPUTs and in Monitor.
4. Backspacing into the word EDIT before typing the line number may disable Ctrl-R (RESTART) in Edit mode.
5. If you inadvertently type the ESCape key before typing Ctrl-E, two warning bells will sound and you will not enter Edit mode. If a line number is typed after the warning bell, hitting Return may delete the line.
6. If a line is longer than the maximum BASIC line length (128 for Integer BASIC, 239 for Applesoft), it will be auto-PACKed, removing all extraneous spaces from the line.

CTRL-W

This is the command to use when you have typed a line in Integer BASIC and received a *** SYNTAX ERROR message. Just type Ctrl-W, and the line will instantly reappear. Ctrl-W works during INPUTs, and it will recover and re-execute immediate commands. This command can also be used to recover lines canceled with Ctrl-X.

Ctrl-W actually works two ways, depending on when it is typed. If it is typed as the first character on a line, it will cause the last line typed from the keyboard to reappear. If it is typed while entering a line, Ctrl-W retypes the line and enters Edit mode.

Here are some things to be careful of when using Ctrl-W:

1. Ctrl-W does NOT work when entered as the first character of an Applesoft line.

2. Very long lines may be auto-PACKed, removing all extraneous spaces.

EDIT MODE: CONTROL COMMANDS

Now that you know how to get into Edit mode, you're ready to try your hand at some Line Editing.

All of the following commands are available only after entering Edit mode as described above. Control characters in the edited line are displayed in Inverse video during Edit mode. A warning bell will sound if your line has reached the maximum BASIC line length (128 for Integer BASIC, 239 for Applesoft).

Notice that the first character of each command's name is the same as (or sounds like) the command itself. This is an easy way to remember these commands.

EDIT MODE COMMANDS

Ctrl-I	INSERT	This command allows you to insert characters into the line. Non-control characters typed after a Ctrl-I will be inserted in front of the cursor, and the rest of the line will spread to make room. All following characters typed will be inserted until a control character is typed (such as <-, ->, or another Edit mode command). To insert control characters, see OVERRIDE.
Ctrl-D	DELETE	To delete the character under the cursor from the line, type Ctrl-D. The character will disappear, and the rest of the line will move in and fill up the space created. If you DELETE too much, you can recover the line with RESTART.
Ctrl-F	FIND	Followed by a character typed N times, FIND will move the cursor to the Nth occurence of the character in the line. For example, to position the cursor on the third colon (:) in a line, type Ctrl-F : : :. Notice that you only need type Ctrl-F once. Type any other character to terminate FIND.
Ctrl-Z	ZAP	This function deletes all characters up to a specific character. A Ctrl-Z followed by a character typed N times will delete all characters up to the Nth

occurence of that character (Much like FIND). For example, to delete three statements from a line, type Ctrl-Z : : :. Type any other character to terminate ZAP. If you ZAP a little too far, you can restore the original line with RESTART.

Ctrl-O OVERRIDE This command functions exactly like the INSERT command except that the FIRST character inserted may be a control character. After the first character has been inserted, OVERRIDE behaves exactly like INSERT, that is, insertion is terminated with a control character. Inserted control characters are displayed in Inverse video in Edit mode. You can use OVERRIDE to insert Ctrl-D's into PRINT statements for DOS commands. You might also want to experiment with other control characters such as the four programmable cursor characters (Ctrl-H,I,J,K), backspaces, and carriage returns.

Ctrl-R RESTART The RESTART command is used to re-edit the original line, regardless of any changes or deletions you might have made (A life saver!). RESTART does not work if you entered Edit mode with a Ctrl-W.

Ctrl-P PACK The PACK command allows you to remove extraneous spaces in edited lines that would otherwise cause a *** TOO LONG error. Ctrl-P will retype a line with all spaces removed except those within quotes. PACK can also be used when you hear the bell warning you that your line is too long. To protect REMark lines from being PACKed, enter a quote as the first character of the REMark.

Ctrl-C CONVERT The CONVERT command is used to quickly convert upper case text to lower case, or vice versa. Ctrl-C converts the character under the cursor to the current case set by the Ctrl-A/Ctrl-S upper-lower shift lock, and advances the cursor.

Ctrl-B BEGIN This command moves the cursor to the beginning of the line. It is useful for repositioning the cursor before using the FIND command, and for renumbering lines.

Ctrl-N	END	This command moves the cursor to the end of the line.
Ctrl-Q	QUIT	The QUIT command deletes to the end of the line starting from the cursor position and exits Edit mode. The line is entered up to the cursor position.
Ctrl-M	RETURN	Typing a Carriage Return will exit Edit mode and save the line exactly as it appears on the screen. This is the usual way of leaving Edit mode.
Ctrl-X	CANCEL	Ctrl-X is used to cancel the line being edited, and exit Edit mode. A line cancelled with Ctrl-X can be recovered with Ctrl-W.
<-, ->		The backward and forward arrow keys work just as they normally do, moving the cursor backward or forward one space. These are also invoked with Ctrl-H and Ctrl-U, respectively.
Ctrl-A, Ctrl-S		The upper and lower shift lock keys also work in Edit mode. Ctrl-A is upper shift lock, and Ctrl-S is the lower shift lock. (See CONVERT)

SECTION IV. LISTING CONTROL

STOP LIST ESC or Ctrl-S	To temporarily stop character output such as program listings or CATALOGs, hit either Ctrl-S or ESC. Output may be resumed by hitting any key.
CANCEL LIST Ctrl-C	To terminate listing entirely and return to BASIC, type a Ctrl-C.
FLUSH LIST Ctrl-F	To quickly skip through long listings, type Ctrl-F. Ctrl-F does not actually stop the listing, it simply disables printing of the characters to the screen. To resume output to screen, type Ctrl-F again.

WARNING: ESC and Ctrl-S have other meanings when typed during character INPUT. The above commands work only when characters are actually being printed on the screen. Typing ESC or Ctrl-S during character INPUT will have unexpected results. If, for example, you are using Ctrl-S to control a program listing, and you inadvertently type Ctrl-S after the listing has ended, you will find yourself typing in lower case. Hitting the ESCape key after the listing has stopped will cause the next character typed to be interpreted as an Escape function Command Keycode. If you type a Ctrl-E in order to edit a line, you will NOT enter Edit mode. Instead, you will hear two warning bells, which means you must type Ctrl-E and try again.

SECTION V. UPPER AND LOWER CASE

Entering lower case text into your Apple is easy with the Program Line Editor. Ctrl-S is the Lower case shift lock key, and Ctrl-A is the Upper case shift lock. Just type Ctrl-S, and all following alphabetic characters typed will be converted and output in lower case. To continue upper case entry, type Ctrl-A. While in Lower case shift lock mode, typing Ctrl-A, Ctrl-X or Carriage RETURN will return you to Upper case mode.

Unless your Apple is equipped with hardware that will display lower case characters as true lower case on the Apple screen (Such as the Paymar Lower Case Adapter), lower case characters entered from the keyboard will be displayed as unintelligible non-alphabetic characters. However, lower case text can be sent to a printer capable of printing lower case characters, even if you don't have a means to display it on th Apple screen. The Ctrl-A/Ctrl-S shift lock commands are available during Edit mode. The Edit mode CONVERT command is handy for quickly converting text from one case to the other.

Here are a few things to keep in mind about lower case input:

1. The Ctrl-A and Ctrl-S characters cannot be entered normally. They must be inserted into a line with the Edit mode OVERRIDE command. You cannot GET a Ctrl-A or Ctrl-S in Applesoft.

2. You can GET lower case characters in Applesoft, but you must ALWAYS PRINT the character after it is input with the GET statement.

SECTION VI. PROGRAMMABLE CURSOR CONTROL

Four control characters, Ctrl-H, I, J and K allow you to control the position of the cursor from within PRINT, REM or INPUT statements. These characters move the cursor left, right, down, and up, respectively. Ctrl-H (Backspace) and Ctrl-J (Line Feed) work even without the Program Line Editor, but Ctrl-I and Ctrl-K require the Line Editor to be up and running to work properly. These Cursor control characters can be inserted into a line with the Edit mode OVERRIDE command. As an example, a PRINT statement with an asterisk followed by five Ctrl-K's and another asterisk will print an asterisk one space ahead and four lines above the other.

Here is a quick summary of the direction of cursor movement when each character is printed:

Ctrl-H - LEFT

Ctrl-I - RIGHT

Ctrl-J - DOWN

CTrl-K - UP

SECTION VII. USING THE ESCAPE FUNCTIONS

Escape functions are User-defineable multi-character commands that are executed with only two keystrokes. An Escape function is invoked by first hitting the ESC key, and then the key that's been assigned to the function. This key is called the Command Keycode for the function. For example, try typing ESC 1. The computer will automatically type CATALOGD1 and a RETURN, and the disk catalog will be displayed. Typing ESC 2 will display the catalog of drive 2.

Escape functions work very much like disk EXEC files, in that characters typed automatically by the Escape function are treated by your Apple exactly as if you had typed them yourself. Escape functions can also be used for cursor movement. In fact, you will notice that all of the Apple's usual ESCape cursor movement commands still work. If you type ESC ->, the cursor will automatically type eight ->'s, quickly copying eight characters from the screen, just as if you had typed them yourself.

The apostrophe character (') can be used to disable the printing of the text of an Escape function to the screen during execution. This is analogous to the DOS NOMON command. For example, ESC W is a handy function that will calculate the start address and length of the most recently BLOADed program (48K systems only). You'll notice, however, that the function itself (a fairly lengthy BASIC PRINT statement) is not actually displayed, just the resultant address and length.

Escape functions can also be defined to be RECURSIVE. After a recursive function has been executed, you can execute another Escape function without having to hit ESC first. This is especially helpful for screen editing and cursor movement. To copy an entire line from the screen, you need only type ESC -> -> -> -> ->, without hitting the ESC key in between. You can use the REPT key with Recursive Escape functions.

ESC I,J,K and M work just like the usual ESC A,B,C and D cursor movement functions except that they are Recursive. Just hit the ESCape key once, and then you can type any combination of the four keys to move the cursor anywhere on the Apple's screen. Notice that the four letters I,J,K and M are arranged in a diamond on the keyboard, with each letter pointing in the direction of cursor movement. (These are the same as the editing features of the Autostart ROM.)

Keep in mind that a character typed after a recursive Escape function will be interpreted as another function command keycode. So, when you want to resume typing, you must terminate the recursive function with a key that does not have a command assigned to it. The space bar is a good choice, since it's not likely to be used for an Escape function command keycode.

NOTE: The Escape functions are not available during Edit mode.

SUMMARY OF SEVERAL INCLUDED ESCAPE FUNCTIONS

Before we see hou you can define your own Escape functions, we'll take a look at some of the ones that have been pre-defined in the standard Program Line Editor program. These will probably give you some ideas for your own Escape functions.

ESC @,A-G — These are the normal Apple Escape editing commands. All work the same as described in your Apple reference manual.

ESC P — Typing ESC P is the same as ESC shift-P, except that you don't have to use the shift key.

ESC I,J,K,M — These recursive functions simulate the Autostart ROM Escape editing funcions. They are the same as ESC A,B,C and D except that ESC need only be hit once.

ESC T — This function types TEXT and RETURN, and then POKE -16300,0 RETURN, which returns you to text page 1.

ESC L — This function types LIST and a carriage return.

ESC 0 — This function automatically types CALL -936.

ESC 1, 2 — ESC 1 will CATALOG the disk in drive 1, and ESC 2 will CATALOG drive 2.

ESC / — This function can be used in place of the word PRINT when writing programs. The "/" is a lower case "?", which is used in place of PRINT in Applesoft.

ESC : — This function does a CALL -151 entry into the Apple Monitor. Note that a ":" is a lower case "*", the Monitor prompt characer. When this function is executed, all that is seen is the word "MON". This is because the function actually first types MON, and then the listing of the function to the screen is disabled. Then, the function types three backspaces and finally CALL -151 and a RETURN.

ESC <-,-> — These recursive functions perform eight <-'s and ->'s, respectively. Typing ESC -> -> -> -> -> will copy an entire line off of the Apple screen, while ESC <- <- will fast-backspace sixteen times.

ESC Q — This function is a handy function that will print the value of the contents of of any two memory locations. To use the function, you must first set the variable A equal to the address of the location of the two-byte number (This is done by typing A=n, where n is an integer). Next, just type ESC Q. This function is roughly equivalent to PRINT PEEK(A)+PEEK(A+1)*256, except that values greater than 32767 are printed as negative numbers. In Integer BASIC, setting A=74 will print the current value of LOMEM:, and A=76 will print the address of HIMEM:. In Applesoft, use A=105 and A=73, respectively, to print the value of LOMEM: and HIMEM:. This function is not printed on the screen during execution.

ESC W — ESC W will calculate the start address and length of the most recently BLOADed disk file on 48K systems. You must change the constants within the function with the ESC CREATE programs for use with other size systems. See page 144 in your DOS 3.2 manual for the constants to use for your size system. This function is not printed on the screen as it is executed.

ESC !-# — ESC shift-1 thru ESC shift-5 make available the five characters that are not normally available on the Apple keyboard: the right bracket (]), the backslash (\), the underline (_), Ctrl-\, and Ctrl-_.

Now that you've seen some of the things that can be done with Escape functions, it is clear that the possibilities are limitless. Escape functions can be used for just about ANYTHING - they can remember those POKEs, PEEKs, and CALLs, type responses to INPUTs, fire up your printer driver, append Integer BASIC programs together... you name it!

CREATING AND MODIFYING THE ESCAPE FUNCTIONS

Each Program Line Editor Greeting program has its own Escape functions imbedded within itself that become available when the disk is booted. There are two programs on the included disk, ESC CREATE.INT and ESC CREATE.FP, that allow you to modify the functions on any Program Line Editor Greeting program. ESC CREATE.INT is used to modify Integer BASIC Greeting programs, and ESC CREATE.FP is used with Applesoft Greeting programs (the .INT or .FP siffixes denote the file type). Neither program will run if the Program Line Editor is not up and running. The operation of the two programs is identical.

To create your own Escape functions, or to modify or delete existing ones, RUN ESC CREATE.INT (Or ESC CREATE.FP, whichever is appropriate for the Greeting program you usually use). In a few seconds, a menu be displayed as shown below:

```
D- DISPLAY ESCAPE FUNCTIONS
E- EDIT/ADD ESCAPE FUNCTIONS
X- DELETE ESCAPE FUNCTIONS
L- LOAD GREETING PROGRAM
S- SAVE GREETING PROGRAM
C- CATALOG
Q- QUIT
```

When the COMMAND prompt appears, type "D". The screen will clear and each Escape funcion will be displayed. Control characters are displayed in Inverse video, just as in Edit mode. Some common control characters are listed at the top of the screen. Note also that a "!" character that appears as the last character of a function's listing indicates that the preceding space is the last character of the function definition. While the functions are being displayed, the ESC or Ctrl-S keys will temporarily stop the listing, and the space bar will abort and return you to command mode (Don't use Ctrl-C - this will return you to BASIC).

To add or edit a particular function, type "E". You will then be asked for the command keycode of the function to be edited or added. If you enter the command keycode for an existing Escape function, (such as "1"), the computer will indicate whether or not the function is recursive, as well as the maximum number of characters that can be contained in the function. The function itself will be displayed with the cursor at the beginning of the function.

If you enter a new command keycode, the computer will prompt you to enter the new function.

In either case, entering or editing Escape Functions is very simple. You can use all of the Edit mode commands except Ctrl-R and Ctrl-X. When you are finished entering or editing the Escape function, hit RETURN to enter it as it appears on the screen (More on the creation of Escape functions later).

Next, you are asked whether or not you want the function to be recursive. If If you decide you don't want the edited function, you can answer this question with an "X" to cancel this function and return to command mode. Any key except "X" or "Y" is considered a "N" response.

Finally, you are asked if you want to change the command keycode of the function. If you answer with a "Y", the computer will prompt you for a new command keycode.

Deleting Escape functions is also very simple. When you type the "X" command, the computer will ask for the command keycode of the function to be deleted.

The L command is used to load a Program Line Editor Greeting program in order to modify its imbedded Escape function table. To load a Line Editor Greeting program, just type "L", followed by the name of the Greeting program you want to load. Next, you will be asked which Escape table you wish to edit. If you respond with an "E", the existing table (the one that became available when the Line Editor was run) will be edited. By responding with an "L", the ESC CREATE program will replace the existing Escape table with the imbedded Escape table of the loaded program.

After you have loaded a Greeting program and edited the Escape function table, you can update the program's imbedded Escape table and SAVE the modified Greeting program to disk with the "S" command. Just type "S", followed by the name of the program you wish to save (not necessarily the same name it was loaded under). You cannot save a program unless one has been previously with the "L" command, however, once a program has been loaded, you can use the "S" command as often as you wish to save copies of the Greeting program.

If at any time you want to try out your functions, just type "Q" to exit to BASIC. To re-enter the ESC CREATE program, type GOTO 100. Do NOT type RUN to restart if you have used the "L" command to load a Greeting program, because the ESC CREATE programs will "forget" that a program has been loaded.

Typing RETURN in response to the COMMAND prompt will re-display the program menu.

ESCAPE FUNCTION APPLICATION NOTES

HOW THEY WORK

When using the ESC CREATE program to display the Escape functions, you've probably noticed that the cursor movement commands, ESC @ through ESC F, are defined in terms of themselves. To see why this is necessary, it is useful to understand how the Escape functions work.

When you hit the ESC key, the Program Line Editor intercepts the ESC character, NOT the Apple's Monitor. The next key typed is also processed by the Line Editor, which searches its own Escape function table for the command keycode. If the function is found, the text of the function is passed on to the Apple Monitor, exactly as if you had typed it yourself. If you want, you can redefine any of the standard ESC @-F cursor movement functions to perform a completely different function. Notice that ESC I,J,K and M are recursively defined with ESC A,B,C and D. This is a sneaky way of simulating the Autostart ROM's ESCape editing features.

If you don't want the text of the Escape function to be displayed on the Apple's screen when it is being executed, just insert an apostrophe (') into the function at the place where you want to disable the display (Usually the first character of the function). When an apostrophe is encountered during the execution of an Escape function, listing of the function is turned off, and the apostrophe skipped over - NOT entered into the line as part of the function.

You can insert ANY character into an Escape function that can be typed from the Apple's keyboard except the apostrophe (') character, which is used to disable output of the function as described above. Carriage returns, <-'s, and ->'s imbedded within an Escape function act just as if they had been typed from the keyboard.

There are a few Escape functions that are intended to prevent accidents when using the Escape functions. Notice that the ESC key itself is defined as a do-nothing, recursive function. This was done so that hitting ESC againg after the execution of another recursive function will not terminate the ESC function. For example, ESC I ESC I will have the same effect as typing ESC I I. The Autostart ROM does not incorporate this feature.

Also notice that ESC Ctrl-E is defined as two bells and a Ctrl-X. This can be a life-saver if you use the ESC key for Stop List, and you accidently hit the ESC key after the listing has stopped. The next key will be processed as an Escape function, and thus you will NOT enter Edit mode. So, if you type Ctrl-E to enter Edit mode and you hear two bells, you know to try again.

SECTION VIII. USING THE LINE EDITOR ON YOUR OWN DISKS

THE PROGRAM LINE EDITOR GREETING PROGRAM

The Program Line Editor is designed to be used as the greeting (HELLO) program on your own disks, thereby providing the Line Editor's editing features automatically when booted.

There are two standard Program Line Editor greeting programs on the supplied disk. One is written in Applesoft, and the other is written in Integer BASIC. These programs can be used as is, or, if desired, they can be modified to provide a custom display, etc. The Escape functions imbedded within any greeting program can be modified at any time with the ESC CREATE programs as described in the previous section.

Here are the names of the standard Program Line Editor greeting programs on the supplied disk:

Integer BASIC - PROGRAM LINE EDITOR

Applesoft - PLE.FP

If you have an Apple][plus system without Integer BASIC, you should have already RENAMEd some of the programs on the Program Line Editor disk. Here are the names of the two standard Program Line Editor greeting programs if you have followed the procedure outlined on page ##:

Applesoft - PROGRAM LINE EDITOR

Integer BASIC - PLE.INT

The easiest way to use the Program Line Editor on your own disks is to use one of the standard greeting programs as the Greeting program on your own disk. It is also possible to rewrite the standard greeting programs. This is easily accomplished, except that there are a few things required of your program in order to allow the Program Line Editor to initialize and run properly. Either way, you should use a Program Line Editor Greeting program that is written in the language that you normally use. Also, don't forget that you can modify the Escape functions on any Program Line Editor Greeting program at any time with the ESC CREATE programs.

USING THE STANDARD PROGRAM LINE EDITOR GREETING PROGRAMS ON YOUR OWN DISKS

Below is a step-by-step procedure for saving a copy of the standard Program Line Editor greeting programs onto your own disks (Commands typed from the keyboard are in CAPS):

a. Insert the Program Line Editor disk and boot normally.

b. Decide which version (Integer BASIC or Applesoft) of the greeting program you want to use.

c. LOAD PROGRAM LINE EDITOR (Or whichever program you have selected)

d. Insert your own disk.

e. CATALOG your disk to find out the name and file type of its existing greeting program. (You may already know this)

f. SAVE HELLO (Or whatever you've named the existing greeting program on disk). You have now replaced the old greeting program with the Program Line Editor. You can save as many copies of your Program Line Editor Greeting program as you wish. See the Note at the end of this section.

WRITING YOUR OWN PROGRAM LINE EDITOR GREETING PROGRAMS

Below is a summary of the procedure to create and save your own Program Line Editor Greeting program:

a. LOAD PROGRAM LINE EDITOR from the supplied diskette.

b. DEL 10,30

c. Enter your program. For an Integer BASIC Greeting program, the last statements executed in your program MUST be the following:

```
GOSUB 32767: PRINT "<ctrl-D>INT"
```

In Applesoft, the last statements executed must be:

```
GOSUB 63999: PRINT "<ctrl-D>FP"
```

Because the final INT or FP immediately terminates the program and clears program memory, it is NOT possible to RUN another program from your Greeting program. BRUNs, CATALOGs, and EXECs, etc., are allowed.

d. Insert your own disk

e. SAVE HELLO (Or whatever you've named the existing greeting program on disk). You have now replaced the old greeting program with the Program Line Editor.

NOTE: If you try to SAVE an Applesoft greeting program onto a disk that has an Integer BASIC greeting program, you will get a FILE TYPE MISMATCH error. To remedy this, you must DELETE the existing greeting program before you save the new one. Be sure to save it under the same name as the old one. It is also possible to INIT a new disk with a Program Line Editor Greeting program.

SECTION IX. CARE AND FEEDING OF THE PROGRAM LINE EDITOR

ENTERING BASIC FROM APPLE MONITOR: CTRL-Y

To return to BASIC after hitting RESET or a CALL -151, type Ctrl-Y and hit return.. If you use 3D0G to reenter BASIC, the Line Editor will be disabled until BASIC is reentered with Ctrl-Y. If you have an Autostart ROM, hitting RESET will always return you to BASIC with the Editor up and running.

RECONNECTING THE EDITOR AFTER USE PR# AND IN# STATEMENTS: CALL 1013 AND &

The Program Line Editor is "connected" to your Apple mush like a printer or other peripheral (PR# EDIT and IN# EDIT, if you will). Therefore, when you use the PR# and IN# statements in your programs, the Line Editor is temporarily disabled and must be reconnected. In Applesoft, this is done with either a CALL 1013 or simply an & in your program or from the keyboard. Since Integer BASIC does not have an & statement, you must use CALL 1013 to reconnect the Line Editor. CALL 1013 and & do exactly the same thing as PRINT"<ctrl-D>PR#0": PRINT"<ctrl-D>IN#0" except that they do not disable the Line Editor. These DOS commands should be replaced by either CALL 1013 or & in your programs.

NOTE: One or both of these capabilities will be disabled if you run a program that uses Ctrl/Y or CALL 1013/& for another purpose. So, it is a good idea to record below the data found by entering the Apple Monitor and typing 3F5.3FA and hitting return. Then, after you have finished with a program that uses either the Monitor Ctrt/Y command or &/CALL 1013, you can reenable them by entering Monitor and typing 3F5:nn nn nn nn nn nn, where nn represents the recorded hexadecimal data.

DATA FOUND AT 3F5:_____

MEMORY CONFLICTS

THE REMOVE PLE PROGRAM

In the unlikely event that a program will not run when the Program Line Editor is in your system, you can BRUN REMOVE PLE on the supplied disk, which will disable the Line Editor and return HIMEM: and the DOS I/O buffers to their normal locations. A memory conflict will usually only occur with very large programs or with other programs that attempt to locate themselves above the DOS I/O buffers (such as Apple Computer's Hi-res character generator program).

NOTE: REMOVE PLE clears program memory when run, and thus cannot be run from within a program.

Here is the procedure for transfering the REMOVE PLE program to another disk:

a. BLOAD REMOVE PLE

b. BSAVE REMOVE PLE,A$300,L$3A

See GENERAL INFORMATION for more information on memory usage by the Program Line Editor.

SECTION X. GENERAL INFORMATION

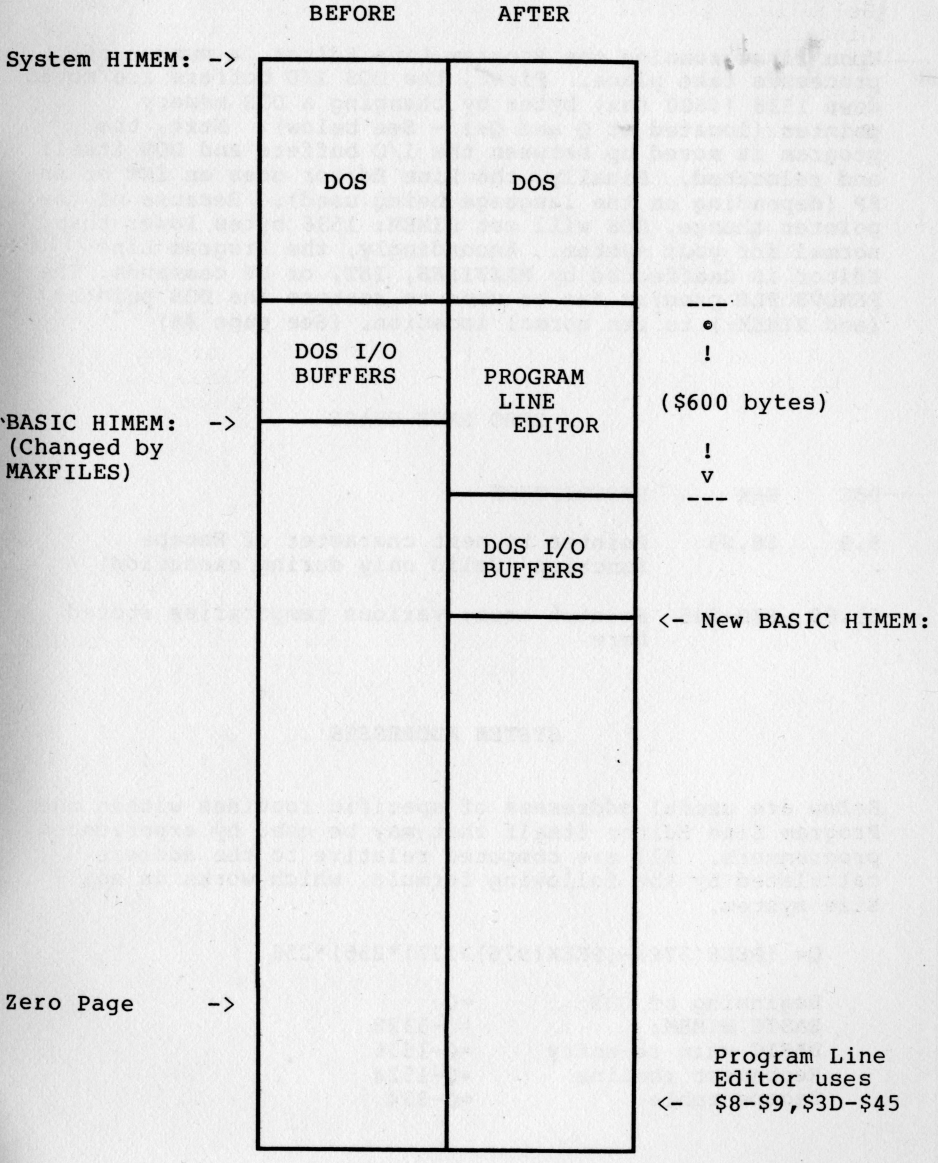
PROGRAMS INCLUDED ON THIS DISKETTE

Below is a list of the programs included on the Program Line Editor diskette. The names will be different on your disk if you have followed the procedure on page ##.

Type	Program Name	Description
I	PROGRAM LINE EDITOR	Integer BASIC Program Line Editor Greeting program.
A	PLE.FP	Applesoft version of above.
I	ESC CREATE.INT	Creates and Edits Escape functions.
A	ESC CREATE.FP	Applesoft version of above.
B	REMOVE PLE	Program to remove the Program Line Editor from the system, freeing up 1536 bytes.

PROGRAM LINE EDITOR INITIALIZATION

MEMORY MAP



THE INITIALIZATION PROCESS

Upon first running the Program Line Editor, a number of processes take place. First, the DOS I/O buffers are moved down 1536 ($600 hex) bytes by changing a DOS memory pointer (located at Q and Q+1 - See below). Next, the program is moved up between the I/O buffers and DOS itself and relocated. Finally, the Line Editor does an INT or an FP (depending on the language being used). Because of the pointer change, DOS will set HIMEM: 1536 bytes lower than normal for your system. Accordingly, the Program Line Editor is unaffected by MAXFILES, INT, or FP commands. The REMOVE PLE program can be used to restore the DOS pointer (and HIMEM:) to its normal location. (See page ##)

ZERO PAGE USAGE

DEC	HEX	DESCRIPTION
8,9	$8,$9	Pointer to next character of Escape function (Valid only during execution)
61,69	$3D-$45	Scratch area; various temporaries stored here

SYSTEM ADDRESSES

Below are useful addresses of specific routines within the Program Line Editor itself that may be used by experienced programmers. All are computed relative to the address calculated by the following formula, which works on any size system.

```
Q= (PEEK(978)-(PEEK(978)>127)*256)*256
```

Beginning of DOS	=Q
BASIC HIMEM:	=Q-3328
BASIC warm re-entry	=Q-1536
Reconnect routine	=Q-1524
Escape table	=Q-334

SYNERGISTIC SOFTWARE
5221 - 120th Ave. S.E.
Bellevue, WA 98006